EXPERIMENT – 9

Name: - Shashwat. Dnyaneshwar Kamdi
Batch – 2 [DevOps Non-Hons]
SAP ID- 500092140
Subject – System Provisioning and Configuration Management Lab

Aim: Creating Multiple EC2 Instances with for each in Terraform.

1] Create a Terraform Configuration File (main.tf)

```
main.tf   instance.tf
terraform-Multiple-EC2-for-each-9 > main.tf > ...
Click here to ask Blackbox to help you code faster | Comment Code |
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.31.0"
    }
  }
}

Comment Code
provider "aws" {
region = "us-east-2"
access_key = "Your IAM access Key"
secret_key = "Your secret access Key"
}
```

2] Create new file name as “instance.tf”

```
main.tf   instance.tf
terraform-Multiple-EC2-for-each-9 > instance.tf > ...
Click here to ask Blackbox to help you code faster | Comment Code |
resource "aws_instance" "ec2_instances" {
  for_each       = var.instances
  ami            = var.instances[each.key].ami
  instance_type  = var.instances[each.key].instance_type
  tags = {
    Name = "EC2-Instance-${each.key}"
  }
}

Comment Code
variable "instances" {
  description = "Map of EC2 instances with settings"
  default = {
    "instance1" = {
      ami           = "ami-05fb0b8c1424f266b"
      instance_type = "t2.micro"
    },
    "instance2" = {
      ami           = "ami-05fb0b8c1424f266b"
      instance_type = "t2.micro"
    },
    "instance3" = {
      ami           = "ami-05fb0b8c1424f266b"
      instance_type = "t2.micro"
    }
  }
}
```

3] Initialize Terraform using command “terraform init”

```
PS F:\UPES\6th Semester\Sys Provisioning and Cnfg Mgmt\Lab\Terraform-Lab-Scripts\terraform-Multiple-EC2-for-each-9> terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.31.0"...
- Installing hashicorp/aws v5.31.0...
- Installed hashicorp/aws v5.31.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

4] Validate it using command “terraform validate”

```
PS F:\UPES\6th Semester\Sys Provisioning and Cnfg Mgmt\Lab\Terraform-Lab-Scripts\terraform-Multiple-EC2-for-each-9> terraform validate
Success! The configuration is valid.
```

5] Check the Plan using command “terraform plan”

```
PS F:\UPES\6th Semester\Sys Provisioning and Cnfg Mgmt\Lab\Terraform-Lab-Scripts\terraform-Multiple-EC2-for-each-9> terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.ec2_instances["instance1"] will be created
  + resource "aws_instance" "ec2_instances" {
      + ami                                  = "ami-05fb0b8c1424f266b"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = (known after apply)
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
```

```
    + tags                                 = {
        + "Name" = "EC2-Instance-instance1"
      }
    + tags_all                             = {
        + "Name" = "EC2-Instance-instance1"
      }
    + tenancy                              = (known after apply)
    + user_data                            = (known after apply)
    + user_data_base64                     = (known after apply)
    + user_data_replace_on_change          = false
    + vpc_security_group_ids               = (known after apply)
  }

# aws_instance.ec2_instances["instance2"] will be created
+ resource "aws_instance" "ec2_instances" {
    + ami                                  = "ami-05fb0b8c1424f266b"
    + arn                                  = (known after apply)
    + associate_public_ip_address          = (known after apply)
    + availability_zone                    = (known after apply)
    + cpu_core_count                       = (known after apply)
    + cpu_threads_per_core                 = (known after apply)
    + disable_api_stop                     = (known after apply)
    + disable_api_termination              = (known after apply)
    + ebs_optimized                        = (known after apply)
    + get_password_data                    = false
    + host_id                              = (known after apply)
    + host_resource_group_arn              = (known after apply)
    + iam_instance_profile                 = (known after apply)
    + id                                   = (known after apply)
    + instance_initiated_shutdown_behavior = (known after apply)
    + instance_lifecycle                   = (known after apply)
    + instance_state                       = (known after apply)
    + instance_type                        = "t2.micro"
    + ipv6_address_count                   = (known after apply)
    + ipv6_addresses                       = (known after apply)
    + key_name                             = (known after apply)
    + monitoring                           = (known after apply)
    + outpost_arn                          = (known after apply)
    + password_data                        = (known after apply)
    + placement_group                      = (known after apply)
    + placement_partition_number           = (known after apply)
    + primary_network_interface_id         = (known after apply)
    + private_dns                          = (known after apply)
    + private_ip                           = (known after apply)
    + public_dns                           = (known after apply)
```

```
    + public_ip                            = (known after apply)
    + secondary_private_ips                = (known after apply)
    + security_groups                      = (known after apply)
    + source_dest_check                    = true
    + spot_instance_request_id             = (known after apply)
    + subnet_id                            = (known after apply)
    + tags                                 = {
        + "Name" = "EC2-Instance-instance2"
      }
    + tags_all                             = {
        + "Name" = "EC2-Instance-instance2"
      }
    + tenancy                              = (known after apply)
    + user_data                            = (known after apply)
    + user_data_base64                     = (known after apply)
    + user_data_replace_on_change          = false
    + vpc_security_group_ids               = (known after apply)
  }

# aws_instance.ec2_instances["instance3"] will be created
+ resource "aws_instance" "ec2_instances" {
    + ami                                  = "ami-05fb0b8c1424f266b"
    + arn                                  = (known after apply)
    + associate_public_ip_address          = (known after apply)
    + availability_zone                    = (known after apply)
    + cpu_core_count                       = (known after apply)
    + cpu_threads_per_core                 = (known after apply)
    + disable_api_stop                     = (known after apply)
    + disable_api_termination              = (known after apply)
    + ebs_optimized                        = (known after apply)
    + get_password_data                    = false
    + host_id                              = (known after apply)
    + host_resource_group_arn              = (known after apply)
    + iam_instance_profile                 = (known after apply)
    + id                                   = (known after apply)
    + instance_initiated_shutdown_behavior = (known after apply)
    + instance_lifecycle                   = (known after apply)
    + instance_state                       = (known after apply)
    + instance_type                        = "t2.micro"
    + ipv6_address_count                   = (known after apply)
    + ipv6_addresses                       = (known after apply)
    + key_name                             = (known after apply)
    + monitoring                           = (known after apply)
    + outpost_arn                          = (known after apply)
    + password_data                        = (known after apply)
    + placement_group                      = (known after apply)
    + placement_partition_number           = (known after apply)
    + primary_network_interface_id         = (known after apply)
    + private_dns                          = (known after apply)
```

```
    + public_ip                            = (known after apply)
    + secondary_private_ips                = (known after apply)
    + security_groups                      = (known after apply)
    + source_dest_check                    = true
    + spot_instance_request_id             = (known after apply)
    + subnet_id                            = (known after apply)
    + tags                                 = {
        + "Name" = "EC2-Instance-instance3"
      }
    + tags_all                             = {
        + "Name" = "EC2-Instance-instance3"
      }
    + tenancy                              = (known after apply)
    + user_data                            = (known after apply)
    + user_data_base64                     = (known after apply)
    + user_data_replace_on_change          = false
    + vpc_security_group_ids               = (known after apply)
  }

Plan: 3 to add, 0 to change, 0 to destroy.
```